TITLE OF THE INVENTION

VIDEO SPLITTING AND DISTRIBUTED PLACEMENT SCHEME FOR CLUSTERED VIDEO SERVERS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of Chinese Patent Application No. 03118543.6, filed on January 25, 2003, in the Chinese Intellectual Property Office, the disclosure of which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0002] The present invention relates to a method of information processing and storage. It involves the fields of parallel and distributed processing and video processing technologies, and a distributed storage method of the clips of source files suitable for cluster video servers.

2. Description of the Related Art

[0003] With the development of the broadband network, audio and video encoding technologies, streaming media and rich media are more and more widely used. For these types of applications, one of the most important infrastructures is the video server. In order to meet the practical needs, the video server has high requirements for performance. The streaming media services are always characterized by large data volume and strict real-time requirements. The general single server architecture can only serve dozens of clients due to the bottleneck caused by a CPU, a memory, a network and a hard disk, and is not applicable for streaming media services that are intended for serving a large number of users, while the high performance servers are very expensive.

[0004] Due to the high scalability and low cost of clusters, the cluster-based technologies provide a technical foundation for the implementation of video servers. Remarkable characteristics of clusters include the decentralization of storage units, the autonomy of individual nodes and centralized control. Based on these characteristics, the cluster video servers have attracted more and more attention in academic communities and the industry.

[0005] One of the key technologies of the cluster video server is to realize the distributed storage of the films. Each movie needs to be split according to some specific methods so that it can be distributed among the storage nodes of a cluster. Hence, it is very important to find a distributed storage method with high efficiency and high availability for the splitting of movies.

[0006] There are two typical solutions for the storage of movies. One is based on the playtime of the movies: dividing the movie into several parts with the same time length (P. Shenoy, P. Goyal, and H. M. Vin, "Issues in Multimedia Server Design", ACM Computing Survey, Vol. 27, No. 4, pp.636-639, December 1995). The other is based on the size of the movies: dividing the movie into several parts with the same file size (P. Shenoy and H. M. Vin. "Efficient Striping Techniques for Multimedia File Server", Proceedings of the 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV' 97), pp. 25-36, May 1997).

[0007] The former one is based on the same time length and the latter is based on the same space. Both of these methods have some problems. It is difficult to realize the splitting strategy based upon the same time. There are two reasons for this. On the one hand, both the compression ratio and storage formats of each media format are totally different. On the other hand, in the files of the same media format, the scene changes among different phases are not predictable so that the storage space of the media data with the same time length is different. Furthermore, this method is not general enough to process completely different media formats. For the splitting strategy based upon one same space size, random access cannot be easily processed. Because it is possible that different media streams in one movie will not be synchronized in some clips, quality problems will result when playing back the movie. Additionally, this method needs synchronous processes of two consecutive clips across two storage nodes. This process decreases the performance of the system processing multiple streams simultaneously. On the other hand, it increases the difficulties of implementing the interface module of the media data in the video servers and the internal communication traffic between control nodes and data nodes.

[0008] The implementation of the two methods described above generally includes four modules: one client information obtaining module, one clip files definition module, one module of obtaining streaming media files information and one media files splitting module. The meanings of these modules in the two methods are distinguished. Consider the client information obtaining module, its main function is to obtain the splitting requirements of the clients. In the splitting method based upon time, it obtains the time length of each clip. While in the splitting method based upon space, it obtains the space length of each clip.

SUMMARY OF THE INVENTION

[0009] The present invention provides a novel video splitting and allocating scheme for clustered video servers.

[0010] In the new video splitting and allocating scheme, both the time-length issue and the space-length issue of each video slice are considered. Simultaneously, distributed allocating schemes to set splintered video slices to different server nodes of clustered video servers, which would be useful for effectively utilizing the parallel processing characteristic are used.

[0011] In one aspect of the present invention, there is provided a video splitting and allocating scheme for clustered video servers.

[0012] 1. The video splitting and allocation scheme for clustered video servers is described below.

Define a structure of a network packet, a structure of a distributed control file, and a structure of a clip file.

Analyze the basic information of a streaming media source file, and process a client requests to obtain a basic splitting requirement defined by the clients request.

Define a split files placement strategy and analyze a clip file allocating requirements defined by the clients.

According to the requirements of the clients, analyze the streaming media source files to construct a splitting task list and the relevant control files.

Create several threads to split the streaming media source files. Each thread is responsible for splitting a media source file.

According to the clip placement strategy, distribute the splintered clip files to the relevant storage server nodes.

[0013] 2. The above video splitting and allocation scheme for clustered video servers has the following characteristics:

The structure of the network packet for the media streaming service, which complies with the form of the main part of the streaming media data message in the international real-time transmission protocol, including the media type header, serial number, time stamp, synchronous signal and main media data.

The streaming media source file information capture procedure includes Index file and SDP file. The Index file includes the transmitting task list, the file name of the source video, the storage space of the source video, the time length of the source video, the clip file number of the source video, and the hot spot of the source video. The SDP file includes the media type, the number of streams included in the source video, the time length of the source video and the ID of the streaming session.

The procedure for defining the structure of clip files, includes the header of clip files, the information header of media streams and the network packet of the media streaming service.

The course for analyzing the information of media source file, analyzes the number of logical time units in the media source files, and obtains the time information of the header and the number of the media stream for each logic time unit. It loops until all the logic time units are finished to get the total playback duration, the storage space of the media file, and the ID of the media type based on the defined structure of the clip file.

The procedure for analyzing client requirement information includes obtaining and analyzing the splitting time requirements and the clip placement strategy.

The clip placement strategy includes the option of data placement strategy, the hot level of the source video and the algorithm for allocating clips to storage server nodes.

The streaming media file analyzing and splitting task list producing procedures capture the clip file placement requirements defined by clients. Meanwhile, the media source file is analyzed to find the space and time deviation of each clip file and the range of serial number of the network packet. Based on this analysis, the splitting task list is produced.

The procedure for splitting the source video first reads the Index file to get the number of clips, and then creates several threads according to this quantity. After this, it reads the Index file to get the play task list, and sends each item in the list to relevant threads so that the splitting task can be created. For each splitting task, locate it to the defined location in the source video file to find pack mark. Then read the data unit when meeting this mark, and according to the description of network streaming media packet defining module, cut the data unit into several packages of network streaming media and write the data unit in a relevant clip file. This process is repeated until all the relevant data units are finished.

[0014] Overall, this invention comprehensively uses the analyzing technology for the standard media formats, the splitting technology for media streams, and real-time transmission protocol realization technology for media streams. It has the advantages of the traditional methods upon time and space and effectively uses the power of the cluster systems. The detailed characteristics of the invention are as follows:

1) High generality

[0015] Theoretically, the system can support all the current media forms, and it does not depend on some specific media form. Although different media forms have different coding standards and storage forms, they have a same character. That is, they organize the media data upon the time index and are stored in the medium. This invention designs an overlay packet structure to packing the different media data.

2) High availability

[0016] The storage system for the split-distributed media data issued in the invention is stable and reliable. It exchanges with the read and write interface in the upper layer of the video server with the sole form standard. In this way, the file W/R interface becomes simple and clear, the fault rate of the whole server is obviously decreased and the reliability of the whole server is increased.

3) High efficiency

[0017] The split distributed storage technology makes video servers highly efficient. Before all the films enter into the storage system, they must accept the file split process of this invention. The split files are stored in several nodes in this split distributed way. The file split process is highly efficient and stable and increased speed can be guaranteed.

4) Significantly increase the quality of clustered video servers

[0018] Through files split, the invention provides the single standard structure for all the stored media files and increases the efficiency of the media data. During the course of reading the files to sending them to the network of the media data, the times of transaction caching decreases 1/3, and therefore saves the system resources of the server nodes. Meanwhile, the size of data packets is almost the same, and the network bandwidth can be used rationally. There will not be sharp changes of bandwidth use so that the network resources are not wasted.

5) Benefit for the realization of the distributed record function of the video server

[0019] As the interactive server, the video must have record function. This function is widely used for remote education, live report and so on. The technical difficulty to realize the distributed record function is how to store the recorded media data in several distributed nodes, while supporting the real-time command of the clients online. The sole standard form of media files designed in the invention is beneficial in the distributed storage of the data distributed random visit. Accordingly, it can overcome the above difficulties of the distributed record function.

6) Benefit for the design of client player for media streaming service.

[0020] The serious difficulty for designing the client player of streaming media is in how to organize the network streaming media packets. There are several streams in the media packet, so organizing the messages, and synchronizing several streams after receiving data in the client present technical difficulties. The structure of the streaming media network packet in the invention saves the synchronization process. Accordingly, it can combine several streams into a single stream for transmitting. So the organization of the packet is relatively simple.

[0021] According to another aspect of the present invention, a computer readable medium is provided, encoded with processing instructions for performing a method of splitting and allocating streaming media source files, the method including: defining a structure of a network packet, a structure of a distributed control file, and a structure of a clip file; analyzing information of streaming media source files, and processing a client's requirement to obtain a splitting requirement of the streaming media source files into clip files; defining a split files placement strategy and analyzing the clip file allocating requirements, according to the client's requirements; analyzing the streaming media source files to construct a splitting task list and relevant control files, according to the clients requirements; creating several threads to split the streaming media source files, wherein each thread is responsible for splitting a streaming media source file; and distributing the clip files to relevant storage server nodes, according to the split files placement strategy.

[0022] Additional aspects and/or advantages of the invention will be set forth in part in the description which follows and, in part, will be obvious from the description, or may be learned by practice of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] These and/or other aspects and advantages of the invention will become apparent and more readily appreciated from the following description of the embodiments, taken in conjunction with the accompanying drawings of which:

FIG. 1 illustrates a flowchart of the process of the invention;

FIG. 2 illustrates an apparatus for performing the method of the present invention;

FIG. 3 illustrates the structure of the network data message for media streaming service;

FIG. 4 illustrates the Index files structure;

FIG. 5 illustrates the SDP files structure;

FIG. 6 illustrates the clip files structure;

FIG. 7 illustrates the data structure of the header information data for clip files;

FIG. 8 illustrates the data structure of the information header for media streams;

FIG. 9 illustrates the data structure of the media stream package in clip files;

FIG. 10 illustrates the process framework for analyzing the media files and producing the splitting task list;

FIG. 11 illustrates the data structure of the splitting task list;

FIG. 12 illustrates the processing framework for executing splitting tasks.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0024] Reference will now be made in detail to the embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to the like elements throughout. The embodiments are described below to explain the present invention by referring to the figures.

[0025] FIG. 1 illustrates a flowchart of the process of the invention. The system first defines the necessary information like split file structure, streaming media network packet structure 12, distributed control file structure of media data 13, definition of clips 14, and split distributed placement strategy. Then the streaming media file information capture module captures the basic information of the source files 15, captures the client's requests information 16, and prepares for the following split. The system then defines the data placement strategy 17, analyzes the streaming media file to produce splitting task list and process the task list 18, splits the files into clips 19, and transmits and stores the clips 20.

[0026] According to the flowchart of the novel media source file splitting scheme, a detailed description of media source files splitting process is described below.

[0027] The media source files splitting process first uses a stream media network message defining procedure to define the network messages. Based on the network packet structure, the streaming media distributed control file defining procedure and split file defining procedure are simultaneously processed defining the structure of the media disconnected control information files and the whole structure of the split files. Then, a media source file information capture module is used to get the basic information of the media source files in preparation for the following work. Meanwhile, the basic split requests of clients are accepted. The requests of clients are chosen according to some key parameters. There are two ways. One is that the clients can define how many clip files according to the quantity of splitting. The other is to regulate the playing time of the clip time to obtain the whole playing time of the media files. Thereafter, the process provides the clip file placement strategies defining procedure to get the placement requests of clip files defined by the clients, and processes the streaming media files analyzing procedure and splitting task lists producing procedure to obtain the splitting task and relevant control information files.

[0028] Next, the split files placement strategy defining procedure is executed to obtain the split files placement request defined by clients. Then the streaming media files analyze and splitting task list procedures are executed to get the splitting task list and relevant control information files.

[0029] Then the splitting task process procedure is executed. Several threads are created. Each thread executes a splitting task. This splitting task first reads the information in the splitting task data structure. Then opens the files and locates the space deviation place referred in the splitting task structure. Then the pack of media data is read. Accordingly, in the streaming media network packet structure defining module, several streaming media network packets will be produced. This procedure is also called RTP packet, which repeats the procedure until all the relevant packs are read. Then the split files are produced.

[0030] Finally, the split transmission storage procedure is executed. Each split file will be stored in the storage node of a relevant cluster upon meeting the requirement of split files placement strategy. So a splitting task is completed. In this splitting task process, the complete percentage of splitting is detected and shown to the client. And the dispatch time information is record so that the client can judge if the procedure is successful or not.

[0031] FIG. 2 illustrates an apparatus for performing the method of the present invention. The apparatus includes various modules including a module to capture client's requests 21, module to define files 22, module to capture the basic information of streaming media source file 23, module to analyze the streaming media files and to create task lists 24, a module to split files into clips 25, and a module to transmit and store clips 26. The module to capture client's requests, includes a module to request and capture the data placement strategy media files and to create a task lists 211, and a module to request and capture the data placement strategy 212. The module to define files 22 includes a module to define streaming media network packets 221, a module to define distributed control files 222, a module to define clips placement strategy 223 and a module to define clips 224.

The module of defining streaming media network packets

[0032] This procedure defines the structure of streaming network packet. In this system, the basic structure of the streaming network packet complies with the detailed standard of streaming media protocol. There are several international standard protocols of streaming media in the field. RTSP (real-time stream protocol) is used for order interaction control between the client and server. RTP/RTCP (real-time transmission protocol and real-time control protocol) are used for regulating and controlling network streaming media packet. SDP (session description protocol) is used for describing the connection between client and server. This system defines the special form of the main part of streaming media packet complied with international RTP protocol. Referring to FIG. 3, a decoding unit is a data unit which can be received and decoded by a decoder. The space size of the unit is generally fixed. The size is marked as Decoding Unit Size. Take MPEG-1 as an example, the size is about 2000 bytes and varies according to different compressing ratios. The system fixes the size of the packet, marked as Packet_Rayload_Size. Now, the key point of the problem is to cut each decoding unit into several packets. Obviously, the size of most packets still relies on the regulated fix size of packets. The size of the remaining packets depends on the size of the remaining data after the decoding unit is cut. Because the decoding unit has a fixed size, the packet is fixed too.

[0033] After the decoding unit is cut, it should be organized in the client. This procedure provides information necessary to guarantee that RTP standard protocol regulates the header of the protocol packet to have a time stamp, a serial number and a media mark. The procedure regulates that several RTP packets in the same decoding unit have the same time stamp, but the serial number should be increased progressively according to the original data moving. Refer to FIG. 3. The relevant packet can be easily organized to a complete decoding unit to playback.

Streaming media distributed control files defining procedure

[0034] A film source file will be cut into several split files and stored in several nodes. When a client requests to play some part of the film, guaranteeing that the video server dispatch all splits of the film without omitting any part is very important. This requires the storage system to provide efficient film split control information.

[0035] In this procedure, the basic information such as playing length serial number, storage location, needed bandwidth will be retrieved in a fixed way. The system regulates that the playing of each split is a basic playing task. Therefore, in the playing of a complete film, there will be a playing task list. The basic playing task list is read to the memory by the video server, then the dispatch can be realized correctly.

[0036] FIG. 4 illustrates the structure of the Index file. There are two distributed control files involved in this structure. One is the Index file, the other is the SDP file. The Index file is constituted by a playing task list, film resource file name, film resource space size, film resource time length, and film resource split quantity and film resource hot spot. The playing task list includes all of the playing tasks. Each playing task is constituted by the start time, finish time, start serial number, finish serial number and IP address of the node machine of the split file of the task.

[0037] FIG. 5 illustrates the structure of the SDP information file. The SDP information file is used to describe the basic information of the film resource and preparation information before the decoding of the client. The SDP information file is constituted by media type number, the quantity of the media streams in the film resource, the time length of the film source and the only mark of the client session.

The module of clips definitions

[0038] The key file of this procedure is the split file. The structure of the split file has an individual standard. According to this standard, the split process module of the system can cut and match the files with any media forms correctly, so that the system can operate without specific media form.

[0039] According to the design of this invention, the split file is constituted by the following parts:

1) Split file header

2) Message header of the media stream

3) Packet of the media stream

[0040] Among the above parts, 2) and 3) emerge in pairs. If there are two streams in this split file, it includes two pairs of 2) and 3). The logic structure of the split files is described in FIG 6.

[0041] The detailed structures of these three parts are as follows.

[0042] 1) Split file header

[0043] FIG. 7 illustrates the data structure of the header information data for clip files. The split file header describes the basic information of the split. It is constituted by the serial number, the time length, the quantity of the media streams, the average network bandwidth and the version number of the split. The average network bandwidth is used to estimate the utilization rate of the network bandwidth after the video server is powered on, so that the bandwidth can be distributed.

2) Message header of the media stream

[0044] FIG. 8 illustrates the data structure of the information header for media streams. The message header of the media stream describes the basic information of the media stream. The media stream refers to the video stream, audio stream or system stream. The video stream with a different coding standard is referred to as a different media stream. This part is constituted by the following information: mark of the media stream (used to distinguish the decoder), playback time length of the media stream, the compressing ratio of the media stream and the data start location (to provide the location of the media data for the reading interface).

3) Packets of streaming media data

[0045] FIG. 9 illustrates the data structure of the media stream package in clip files. In the definition module of streaming data packet, the structure of the streaming media packet has been described in detail. FIG. 3 shows that one sequence number is an unsigned integer. Because the sequence number is in a limited range, it is possible that the value of the sequence number reverses. The value of the time stamp is an unsigned integer with 32 bits, and the time stamp is the product of the time and the bit rate of the media stream. When there is a request with random accessing operations, it will fail since the values of the sequence numbers and time stamps have the possibility of reversing. It is very necessary that the packets of streaming media be encapsulated, including the IDs of media streams, the packet sequence number and the real playing back point (count from zero). With the help of these three items, it is very easy to randomly access the streaming media data.

The module of getting the basic information from source media files

[0046] All data structures are defined in the method. The objective of the module is to capture the basic information of the source media files, including space size, the number of media streams, the time length of each media stream, the media format ID of each media stream and the total time length of the source file. Since the specifics of each media stream are different, it is necessary to program a different module for different media streams.

[0047] The MPEG-1 system stream will be described as an example of how to capture the basic information of source media files. The procedure is as follows:

[0048] In a first operation, the file structure of the MPEG-1 media format has a strict definition. There is a system header in the source media file. The media format ID, the compression ratio and other information can be captured from the system header.

[0049] In a second operation, the media data should be analyzed. The media data of MPEG-1 are organized into many Packs. The size of each pack is fixed. Hence it is an individual decoding unit involved above. Each pack has a fixed message header, which records the playtime of the pack. Therefore from the beginning to the end, until the header of the last pack is analyzed, the whole playtime of the source media file can be captured.

[0050] With these two operations, the work to capture the information of media source files has been accomplished.

The module to define the clip files placement strategy

[0051] The clip file should be stored on all the storage nodes according to some rules and strategies. When the placement strategy is being designed, the high visiting frequency of the hot films, the load balance of each storage node and the backup of the system should be considered. The module includes a data placement strategy option, hot level options of films and the data placement algorithm.

[0052] The system provides a data placement strategy with client control. Firstly, a typical data placement strategy is provided: round robin; secondly, hot level options of films are given by clients and can be used to decide the replicas of each clip of one film. The system can finish the distributed storage of all clip files of films according to the above information provided by clients.

[0053] The hot level option is defined as Hot. Data structures are defined as follows:

```
Hot_Level {

First_Level;   // level 1. All the clips have no replicas;

Second_Level; // level 2. Each clip of the media file has a replica.

Third_Level; // level 3. Each clip of the media file has two replicas.

Top_Level; //the top level. Each clip of the media file has three replicas.

};
```

[0054] The algorithm of round robin strategy is as the follows:

[0055] Given N storage nodes: Host[I], I=1,2,......N,;

[0056] All the M clips of one film are: Clips[I], I=1,2,......M.

[0057] The first replicas set is defined: Clip_One[I], I=1,2,......M;

[0058] The second replicas set is Clip_two[I], I=1,2,......M;

[0059] The third replicas set is Clip_three[I], I=1,2,......M.

[0060] When the hot level option is First_Level,

[0061] The storage location of the I-th clip is Host[a]: Host[a]=I mod N;

[0062] When the hot level parameter is Second_Level,

[0063] The storage location of the I-th clip is Host[a]: Host[a]=I mod N;

[0064] The storage location of the J-th replica clip in the first replicas set is Host[b]:

[0065] Host[b]=(J mod N)+1;

[0066] When the hot level parameter is Third_Level,

[0067] The storage location of the I-th clip is Host[a]: Host[a]=I mod N;

[0068] The storage location of the J-th replica clip in the first replicas set is

[0069] Host[b]: Host[b]=(J mod N)+1;

[0070] The storage location of the K-th replica clip in the second replicas set is

[0071] Host[c]: Host[c]=(K mod N)+2;

[0072] When the hot level parameter is Top_Level,

[0073] The storage location of the I-th clip is Host[a]: Host[a]=I mod N;

[0074] The storage location of the J-th replica clip in the first replicas set is

[0075] Host[b]: Host[b]=(J mod N)+1;

[0076] The storage location of the K-th replica clip in the second replicas set is

[0077] Host[c]: Host[c]=(K mod N)+2;

[0078] The storage location of the L-th replica clip in the third replicas set is

[0079] Host[d]: Host[d]=(L mod N)+3;

The module to analyze the Streaming media files and to create task lists

[0080] The module of analyzing the Streaming media files is responsible for capturing information of files and is ready to create the task lists of splitting files into clips. The data placement strategy can be parsed from the requests of clients. Meanwhile the media source files should be analyzed to find the space interval points, time interval points and the sequence number range of network packets of each clip. Then the task of how to split files into clips can be created. The procedure is described as follows:

[0081] A first operation is performed to search the defined PACK mark according to the captured basic information of media files.

[0082] In a second operation, the data of the PACK units should be processed to analyze the time stamp and the sequence number.

[0083] In a third operation, the time stamp should be compared with the end time of each clip file. If equal, a clip file list item can be written into the Index files. Otherwise, the next PACK data unit will be handled again.

[0084] FIG. 10 illustrates the process for analyzing the media files and producing the splitting task. In a first operation, the load source media files are loaded 100, thereafter, a determination is made as to whether the file ends 101. If the file ends, the clip placement strategy file information is loaded 102, and each clip is customized according to all the pack information 103. At a later operation, a location, start and end pack number and time in the splitting task item are written 104, the capture task list is created 105 and the process ends. If the file has not ended, media stream and buffer information are read 106, and a PACK header information is searched for 107. Thereafter, a determination is made as to whether a PACK mark is found or not 108. If no PACK mark is found, the process returns to operation 101. If a PACK mark is found, the PACK data is analyzed and the time is counted 109. Thereafter, the media time is transferred into the network time 110, and the size and mark of one pack are captured 111. Finally, the above information of the PACK is recorded 112.

[0085] FIG. 11 illustrates the data structure of the splitting task list. From the figure, the task lists record the basic information of each clip file, including the space offset in the source files, start time point, end time point and the flag whether the process is successful or not.

The module to split files into clips

[0086] In this procedure, multi-threads are created to split the source files into clips according to the above task lists. The procedure is described as follows:

[0087] In a first operation, the procedure will read the Index files to get the number of clips of one source file and then create multiple threads according to the clips number.

[0088] In a second operation, the procedure will continue to read the Index files to get the task lists, and transmit each item in the lists to relevant threads to establish each splitting task.

[0089] For any splitting task, a third operation will analyze the source files and locate the location of PACK marks. When meeting the mark, the PACK will be read out and be split into several network packets according to the module to define the data structures of network packets. Then all network packets will be written into relevant clip files.

[0090] In a fourth operation, the procedure will repeat the process of the third operation until all of the corresponding data units are handled over, and all splitting work has been done.

[0091] In the process of each splitting task, the percentage of the splitting work will be showed to the clients. And the scheduling time information will be recorded for the clients to judge it as successful or not.

[0092] FIG. 12 illustrates the processing framework for executing the splitting tasks. In a first operation, the splitting task lists are loaded 120, thereafter the source media files are loaded 121, and a multi-threads are created according to the splitting task lists and all the splitting tasks are processed 122. For each splitting task, a corresponding splitting task item is read, including the location pack number and a start time and end time 123. For each splitting task, located in the relevant source file 124. Thereafter, a determination is made at to whether an end location has been reached 125. If no end location has been reached, reach pack is read, analyzed and split into RTP packets 127, and each RTP packet in the clip file is read 128. If it is determined that the end location has been reached, the splitting task ends 126.

The module to transmit and store clips

[0093] This module is used to store the clips created by the above modules on the corresponding storage nodes.

[0094] First, the network addresses of the storage nodes are captured according to the location requests defined in data placement strategy.

[0095] Second, the captured clips are transmitted and stored.

[0096] Although a few embodiments of the present invention have been shown and described, it would be appreciated by those skilled in the art that changes may be made in this embodiment without departing from the principles and spirit of the invention, the scope of which is defined in the claims and their equivalents.